

Leader-Contention-Based User Matching for 802.11 Multiuser MIMO Networks

^{†‡}Tung-Wei Kuo, [†]Kuang-Che Lee, [†]Kate Ching-Ju Lin and [‡]Ming-Jer Tsai

[†]Research Center for Information Technology Innovation, Academia Sinica, Taiwan

[‡]Department of Computer Science, National Tsing Hua University, Taiwan

Abstract—In multiuser MIMO (MU-MIMO) LANs, the achievable throughput of a client depends on who are transmitting concurrently with it. Existing MU-MIMO MAC protocols however enable clients to use the traditional 802.11 contention to contend for concurrent transmission opportunities on the uplink. Such a contention-based protocol not only wastes lots of channel time on multiple rounds of contention, but also fails to maximally deliver the gain of MU-MIMO because users randomly join concurrent transmissions without considering their channel characteristics. To address such inefficiency, this paper introduces **MIMOMate**, a leader-contention-based MU-MIMO MAC protocol that matches clients as concurrent transmitters according to their channel characteristics to maximally deliver the MU-MIMO gain, while ensuring all users to fairly share concurrent transmission opportunities. Furthermore, **MIMOMate** elects the leader of the matched users to contend for transmission opportunities using traditional 802.11 CSMA/CA. It hence requires only a single contention overhead for concurrent streams, and can be compatible with legacy 802.11 devices. A prototype implementation in USRP-N200 shows that **MIMOMate** achieves an average throughput gain of 1.42x and 1.52x over the traditional contention-based protocol for 2-antenna and 3-antenna AP scenarios, respectively, and also provides fairness for clients.

Index Terms—Multiuser MIMO, User matching, Channel orthogonality

I. INTRODUCTION

With the growing technique of multiple antenna systems, the number of antennas on an access point (AP) is increasing steadily. Most of mobile devices, such as smartphones or tablets, are however limited by their size and power constraints, and hence have a fewer number of antennas as compared to the AP. Traditional 802.11 protocols, which enable only a single client to communicate with the AP, hence cannot fully utilize concurrent transmission opportunities supported by a multi-antenna AP. To address this problem, recent work has advocated developing multiuser MIMO (MU-MIMO) LANs [1]–[7] to enable multiple clients to communicate concurrently with an AP and fully utilize all the available *degrees of freedom* [8].

Though some MU-MIMO MAC protocols [1]–[7] have been proposed to realize concurrent transmissions across different nodes, in either the uplink or downlink scenarios, they simply select a random subset of users to communicate concurrently with an AP. However, in a MU-MIMO LAN, the achievable throughput of a client highly depends on who are transmitting concurrently with it. Consider the example in Fig. 1(a), where three single-antenna clients contend for communicating with a 2-antenna AP. Say clients C1 and C3 win the first and

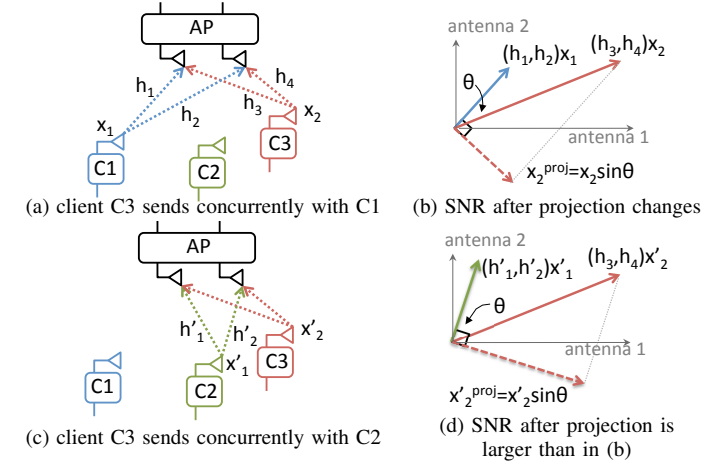


Fig. 1. SNR after projection changes

second contentions, respectively, and transmit concurrently to the AP. The 2-antenna AP receives the signals in a 2-dimensional antenna space, as shown in Fig. 1(b). The basic approach for decoding the concurrent packets is called *zero-forcing with successive interference cancellation* (ZF-SIC) [5] [8]. The AP first zero-forces (ZF) the interference from client C1 by projecting the signal of the second client C3 on a direction orthogonal to C1, and hence can decode C3. The AP then uses interference cancellation (IC) to subtract C3's signal and decode the first client C1. We note that the second client decoded by ZF however experiences SNR reduction after projection, as shown in Fig. 1(b), while the first client decoded by IC obtains the same SNR as it transmits alone. The amount of SNR reduction for the second client depends on channel orthogonality between two concurrent clients, i.e., the angle θ between C1 and C3. Consider another example in Fig. 1(c), where clients C2 and C3 transmit the first and second streams, respectively. Client C3 in this case gets a higher SNR after projection, as in Fig. 1(d), and thus achieves a higher throughput, as compared to sending concurrently with C1, because its channel is more orthogonal to C2's channel. This example illustrates that random user selection cannot efficiently deliver the gain of a MU-MIMO LAN.

To better deliver MU-MIMO gains, some theoretical works on *downlink* MU-MIMO LANs have focused on allowing the single transmitter, i.e., AP, to select a proper subset of users as concurrent receivers. User selection in the *uplink* scenario is however much more challenging because multiple contending clients compete for transmission opportunities without coordination. As a result, existing uplink MU-MIMO LANs [4] [5]

still adopt traditional 802.11 contention to ask all the clients to contend for concurrent transmissions without sophisticated user selection. Such a contention-based scheme not only fails to select concurrent transmitters according to their channel characteristics, but also wastes a lot of channel time for multiple rounds of contention for concurrent transmission opportunities. For example, if multiple single-antenna clients contend for transmitting to a 3-antenna AP, they need to contend for three transmission opportunities sequentially. If a client fails to win the first contention, it needs another round of contention to contend for the opportunity of sending the second stream. If it fails again, it needs to contend for transmitting the third stream. The channel time occupied by such sequential contention could significantly offset the gain of MU-MIMO LANs.

Though some prior studies work on uplink MU-MIMO user scheduling, they either maximize the sum rate of concurrent transmissions without considering fairness [9] [10], or cannot be compatible with legacy 802.11 contention-based MAC [11]–[17]. In addition, all the above proposals do not give any formal model to formulate the relationship between two conflict design goals, i.e., throughput and fairness. Hence, in this paper, we propose *MIMOMate*, a *leader-contention-based* MU-MIMO MAC protocol that matches concurrent transmitters according to their channel characteristics to maximally deliver the MU-MIMO gain under the fairness requirement. Our contributions are as follows:

- We first formulate, in Section IV, a rigorous model to formally define the user selection problem of maximizing throughput under the fairness constraint, and propose a matching algorithm to solve the problem. We further show that *MIMOMate*'s user matching algorithm can achieve the optimal solution for the 2-antenna AP scenario.
- We propose a MU-MIMO MAC design, in Section V, that integrates our proposed user matching algorithm with a *leader-based* contention scheme. *MIMOMate*'s leader-contention-based MAC is hence compatible with traditional 802.11, and, more importantly, requires only a single contention overhead not scaling up with the number of concurrent streams.
- Unlike prior theoretical work that only mathematically analyzes the effect of MU-MIMO user selection, we build a prototype of *MIMOMate* using the USRP-N200 radio platform [18], and use testbed measurements to understand the inefficiency of random user selection in real channels in Section III.
- We finally experimentally evaluate, in Section VI, the performance of *MIMOMate*. The results show that *MIMOMate* achieves an average throughput gain of 1.42x and 1.52x over the sequential-contention-based protocol for 2-antenna and 3-antenna AP scenarios, respectively, and also provides users fair transmission opportunities.

The remainder of this paper is organized as follows. We review related works in Section II. Section III measures how existing schemes fail to deliver MU-MIMO gains and provide fairness in real channels. Sections IV and V describe

our *MIMOMate* algorithm and how to realize it as a MAC protocol, respectively. In Sections VI and VII, we evaluate the performance of *MIMOMate* via experiments and simulations, respectively. Finally, Section VIII concludes this paper.

II. RELATED WORK

In the last few years, the advantage of MU-MIMO LANs has been verified theoretically [19]–[21] and demonstrated empirically [1]–[5], [22], [23]. In Beamforming [1]–[3], a multi-antenna AP uses the precoding technique to transmit multiple streams to multiple single-antenna clients. SAM [4] focuses on the uplink scenario and allows multiple single-antenna clients to communicate concurrently with a multi-antenna AP. TurboRate [5] proposes a rate adaptation protocol for uplink MU-MIMO LANs. IAC [22] connects multiple APs through the Ethernet to form a virtual MIMO node that communicates concurrently with multiple clients. 802.11n⁺ [23] enables concurrent transmissions across different links. All the above practical MU-MIMO systems leverage the traditional 802.11 content mechanism to share concurrent transmission opportunities. In contrast, *MIMOMate* enables clients with a better channel orthogonality to transmit concurrently.

Prior theoretical work on user selection in downlink MU-MIMO LANs [24]–[29] selects the optimal subset of clients from those who have packets queued in the AP to maximize the sum rate of concurrent transmissions. The works [26] [28] further address the issue of fairness. However, their solutions are designed for downlink MU-MIMO, and cannot be easily applied in the uplink scenarios due to the lack of a coordinator.

User selection in uplink MU-MIMO LANs [9] [10] requires the AP to explicitly coordinate between the clients for every packet and select the optimal subset of clients to transmit at the rate specified by the AP. Enabling coordination among clients for uplink traffic however requires a significant signaling overhead. Our work differs from those user selection algorithms in that it matches multiple potential subsets of concurrent transmitters to improve the system throughput, but elects a leader from the matched users to perform traditional 802.11 contention without coordination among clients. Some previous works [4], [11]–[13] propose to use multi-round contention to enable as many concurrent transmissions as possible, while giving clients a fair opportunity to transmit concurrent streams. SAM [4] proposes a preamble-counting protocol, which allows each client to count the number of existing streams and determine whether it can contend for sending a concurrent stream in a distributed way. Multi-round contention [11] [12] is proposed to let clients send RTSs in multiple rounds of contention. Multiple clients might send RTSs concurrently in each single round of contention, and the AP then feedbacks a CTS to notify those clients that can transmit concurrently. An asynchronous MAC protocol [13] is proposed to enable clients to independently start their concurrent transmissions, i.e., without the need of starting concurrent transmissions at the same time. It however relies on a control channel to feedback who can join concurrent transmissions. There are also some papers that have considered channel orthogonality and fairness jointly in uplink MU-MIMO [14]–[17]. However,

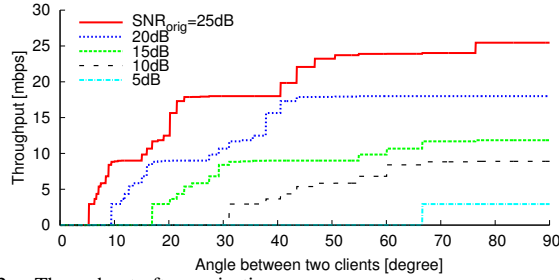


Fig. 2. Throughput after projection

these solutions are heuristics without any formal performance analysis, and also not compatible with the existing 802.11 standard. Our work is the first that maximizes the throughput under the fairness constraint, and is able to coexist with legacy 802.11 devices.

III. MU-MIMO BACKGROUND AND MOTIVATIONS

Before describing our proposed protocol, we first use testbed measurements in real channels to understand the limitation of the existing MU-MIMO MAC protocols. The measurement results also give us an insight to the motivation of enabling user selection in a MU-MIMO MAC. We consider again the network in Fig. 1(a) where two single-antenna clients communicate with a 2-antenna AP. The measurements are empirically performed using USRP-N200 [18] on a 10 MHz OFDM channel with 802.11 modulations and coding rates. The available bit-rates hence range from 3–27 Mb/s. The measurements are designed to answer the following questions:

a) How often is a client unable to transmit concurrently? Recall that the second client's SNR reduces after projection, and the amount of SNR reduction depends on the angle between its channel and the channel of its concurrent transmitter, i.e., the first client. This means that the throughput of the second client in a MU-MIMO network depends on not only its own SNR but also channel orthogonality between concurrent transmitters. To illustrate this point, in Fig. 2, we analytically compute the throughput of the second client decoded by ZF for the whole range of the inter-client angle $\theta \in [0, \pi/2]$ when its original SNR at the AP is 5, 10, 15, 20 and 25 dB, respectively.¹

The figure shows that a small inter-client angle significantly reduces the SNR after projection and hence the throughput. If the client has a high original SNR, e.g., 25 dB, by selecting the best bit rate according to the SNR after projection [5], it can still get a relatively high throughput even after projection. In contrast, if the original SNR of the client is low, the client would be very likely to get zero throughput even if it already uses perfect bit rate adaptation. This is because a small SNR reduction could make its SNR after projection become lower than the 802.11 operational SNR region, i.e., 4 dB. In particular, if the client's original SNR is 10 dB, then for any angle smaller than 31 degree, its SNR after projection drops below 4 dB, leading to zero throughput. The situation is even worse if the client's original SNR is only 5 dB.

¹We empirically measure an SNR-throughput mapping table and map the SNR after projection to the corresponding throughput.

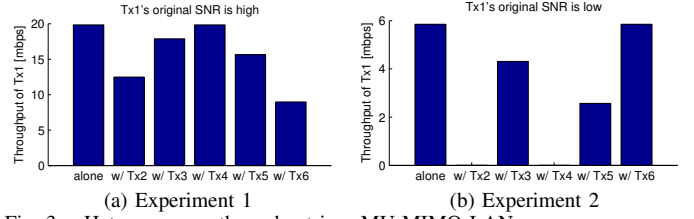


Fig. 3. Heterogeneous throughput in a MU-MIMO LAN

One important thing worth noting is that the success of decoding the first client using ZF-SIC relies on the AP being able to decode the second client correctly. Otherwise, the AP cannot remove the interfering signal of the second client, and hence also fails to decode the first client. Also, after removing the interfering signal from the second client, the first client can select its best bit-rate according to its original SNR without considering who later joins the concurrent transmission. The above constraint hence requires the second client to give up its transmission opportunity if its SNR after projection is lower than the 802.11 operational SNR region. This also motivates why selecting a suitable subset of clients to transmit concurrently is important for delivering the gain of MU-MIMO.

b) How different is the throughput of a client when it transmits concurrently with different clients? We have shown in Fig. 2 that the throughput of a client could change significantly with the inter-client angle. We next check whether the angle between the channels of two clients is actually randomly distributed between $[0, \pi/2]$. To validate this point, we empirically measure how much throughput a client can achieve if it transmits concurrently with different clients in a real testbed where 6 single-antenna clients, named Tx1-Tx6, contend for communicating concurrently with a 2-antenna AP. We repeat the experiment twice with different random locations of the clients. Experiment 1 locates Tx1 close to the AP, while Experiment 2 locates Tx1 far from the AP. Two experiments represent the scenarios when Tx1 has a high and low original SNR, respectively.

Fig. 3 plots the throughput of one client (denoted by Tx1) when it transmits alone or when it transmits concurrently with any of other five clients (denoted by Tx2–Tx6). The figure shows that, in both experiments, as compared to transmitting alone, Tx1 usually gets a lower throughput when it joins the concurrent transmission and is decoded by projection. In addition, the client's throughput, as transmitting concurrently with different users, could be very different. For example, in experiment 1, Tx1 obtains a high throughput when it transmits concurrently with Tx4, while suffering a low throughput as joining Tx6's transmission. The situation becomes worse when Tx1 has a low original SNR (as in experiment 2); in many cases, it gets zero throughput as transmitting concurrently with another client. These results are consistent with the analysis shown in Fig. 2. Thus, to get a high throughput, Tx1 would like to transmit with a client whose channel is more orthogonal to its channel, as a result experiencing less SNR reduction. The current random access protocols however do not consider this effect, and hence cannot efficiently deliver the MU-MIMO gain.

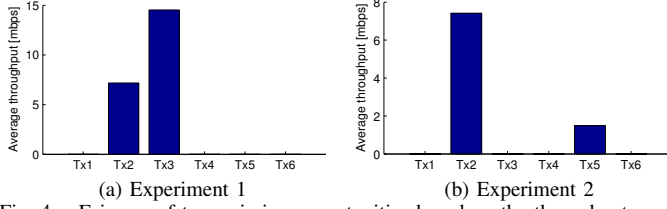


Fig. 4. Fairness of transmission opportunities based on the throughput

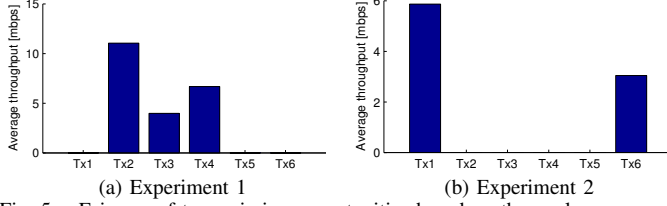


Fig. 5. Fairness of transmission opportunities based on the angle

c) Is selecting concurrent clients to maximize the throughput fair enough? A naïve solution to improving the throughput is to deterministically assign the client that achieves the highest throughput to join the transmission of the first contention winner. This simple solution however might not be fair for all the clients. To see why this is a problem, we measure the throughput of each client in the above scheme. Specifically, in each experiment, we let each client have an equal probability to transmit the first stream; given the first contention winner, we then select the client that can achieve the highest throughput after projection to transmit the second stream. Each experiment includes 1,000 rounds of contention, i.e., 1,000 concurrent transmissions.

Because every client has an equal probability to win the first contention, we do not consider the throughput of a client transmitting using the first stream. Instead, in Fig. 4, we only plot the average throughput of a client transmitting in the second stream. The results of two experiments show that, by applying such a naïve solution, some clients, e.g., Tx1, Tx4, Tx5 and Tx6 in experiment 1, do not have any opportunities to join concurrent transmissions. Even worse, the clients with a low original SNR are very likely to starve because they cannot compete with those clients in the high SNR regime. One might think, alternatively, we can assign the client that has the largest inter-client angle with the first contention winner to transmit concurrently. We repeat the same experiment by applying the above assignment. The results in Fig. 5 show that this solution again fails to provide fairness because some clients happen to have a small angle with all the other clients and hence do not get any transmission opportunities.

IV. MIMOMATE MATCHING

Motivated by the above measurements, we aim at designing a matching protocol, called *MIMOMate*, that pairs concurrent clients to deliver the maximum throughput gain enabled by concurrent transmissions, while, at the same time, providing clients fair concurrent transmission opportunities. For simplicity, we describe our *MIMOMate* protocol assuming that multiple single-antenna clients communicate concurrently with a multi-antenna AP in an uplink MU-MIMO LAN. Our design however can be generalized to clients with multiple antennas and downlink MU-MIMO LANs. We will describe in the next

section how to realize *MIMOMate* as a leader-contention-based MAC protocol.

A. Overview

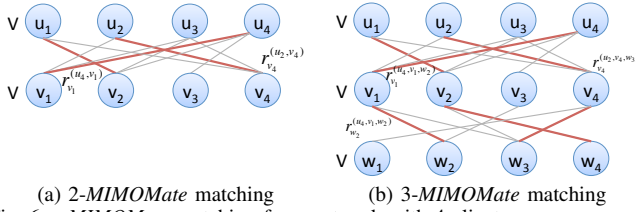
The goal of *MIMOMate* is to build chain relation in concurrent transmissions. When a client wins contention, the following concurrent transmissions are determined a priori. Hence, all the concurrent transmitters only require to precede their streams with one contending process. In particular, *MIMOMate* matches clients whose channels are more orthogonal to each other as a group of concurrent transmitters with a precedence relation, which is called *MIMO-mates*. To see how it works, let us consider an example where two clients are allowed to communicate concurrently with a 2-antenna AP. We match two clients as *MIMO-Mates*, in which one is the lead and the other is the follower. When the lead of *MIMO-Mates* wins contention and transmits the first stream, its follower transmits the second stream concurrently immediately after it detects the transmission from the lead.

The protocol can be generalized to an N -antenna AP scenario where N clients can transmit concurrently. We match N clients as a *MIMO-Mate* (precedence) relation (u_1, u_2, \dots, u_N) such that clients u_1, u_2, \dots, u_N join the concurrent transmissions one after another in order of precedence. In particular, after the lead u_1 wins contention, any of the following clients u_i can count the number of preambles to figure out the time that it should transmit. The above protocol benefits throughput gains from two factors: 1) it matches clients with a higher channel orthogonality to transmit concurrently and minimizes throughput reduction caused by projection; 2) it requires only one contending process for concurrent transmissions, as a result reducing the overhead significantly. However, the benefit of *MIMOMate* might not be able to be fully delivered when any of the matched *MIMO-Mates* does not have traffic to send. In this case, the unused transmission opportunities should be exploited by other clients to avoid waste. We thus further integrate *MIMOMate* with an angle-based contention mechanism, which will be discussed in Section V.

B. Problem Formulation

Our objective is to match clients as *MIMO-mates* in order to maximize the throughput subject to the fairness constraint. We first define our problem in a 2-antenna AP scenario, and next extend it to a 3-antenna AP scenario and even a more general N -antenna AP scenario.

Let us first consider the 2-antenna AP scenario. Say u is the client that wins the first contention, and v is the follower of u , who joins v 's transmission. We define (u, v) as the *MIMO-mate* relation of clients u and v . Let $r_v^{(u,v)}$ denote the throughput of v as it transmits concurrently with u and is decoded by using ZF to project orthogonal to client u . We note that client v might get a different throughput if it is assigned to follow a different predecessor, i.e., $r_v^{(u,v)}$ could be different from $r_v^{(u',v)}$ if $u \neq u'$. The *MIMO-mate* matching problem in a 2-antenna AP scenario can be defined as follows:



(a) 2-MIMOMate matching (b) 3-MIMOMate matching
Fig. 6. MIMOMate matching for a network with 4 clients

Problem 1. (2-MIMOMate) Given a set of clients V and the throughput $r_v^{(u,v)}$ for all $u, v \in V$, the matching problem is to find a set $M \subseteq V \times V$ such that

- 1) $r_v^{(u,v)} > 0, \forall (u, v) \in M$,
- 2) $u \neq v, \forall (u, v) \in M$,
- 3) $u_1 \neq u_2$ and $v_1 \neq v_2$ for any two distinct elements $(u_1, v_1), (u_2, v_2) \in M$,
- 4) $|M|$ is maximized,
- 5) $\sum_{(u,v) \in M} r_v^{(u,v)}$ is maximized among those M satisfying Constraint 4.

To ensure the success of ZF-SIC decoding, Constraint 1 allows a client to join the concurrent transmission only if it can be successively decoded, i.e., getting a positive throughput. Constraints 2–3 force each client to follow at most one of other clients, and hence guarantee fairness. The rationale of Constraint 4 is that, since each client in traditional 802.11 has an equal probability to win the first contention and transmit the first stream, then, by finding the maximum set M , we allow as many clients as possible to join the concurrent transmission. This ensures clients to also have a fair probability to transmit the second stream. Under such a fairness constraint, our goal is to find a feasible solution that maximizes the total throughput of the followers, i.e., the second streams. Note that we are only interested in the throughput of the followers because, by using ZF-SIC, a client who transmits the first stream can get about the same throughput no matter who its follower is [5].

The above 2-MIMOMate problem can actually be illustrated as a bipartite graph, as shown in Figure 6(a). Each edge (u, v) is associated with a weight, which is set to the throughput of v when it follows u , i.e., $r_v^{(u,v)}$. We observe that the 2-MIMOMate problem is exactly equivalent to the *Bipartite Maximum Weighted Maximum Cardinality Matching* problem, which finds a maximum cardinality matching with maximum weight in a bipartite graph and can be solved in polynomial time by the algorithm proposed in [30] (see Chapter 7.8). Note that, since MIMOMate always assigns any leader a specific follower, any feasible solution of Problem 1 is therefore deterministic. That is, the relationship between the first contention winner and its follower is fixed until we solve Problem 1 again when the channels change. This is very different from the probabilistic nature of *uniformly random* contention used in most of existing protocols [4], [11], [12], where every client has the same probability to win the contention of sending concurrently with a given first winner. A natural question then arises: Would choosing the follower uniformly randomly, e.g., via *uniformly random* contention, results in a solution better than the output of Problem 1? We give the following positive result on Problem 1:

Theorem 1. *If the throughput $r_v^{(u,v)}$ is greater than 0 for all*

$u, v \in V, u \neq v$, then the average throughput of Problem 1's output is higher than or equal to the average throughput achieved by using uniformly random contention to choose the follower, even if the contention overhead is ignored.

In fact, we can prove the following stronger theorem, which replaces the *uniformly random* contention by any probabilistic assignment that obeys the fairness constraint. More specifically, in any fair probabilistic assignment, every client has the same probability to transmit the second stream; however, any first winner might choose its follower with a non-uniform probability, and different leaders might have different probability distributions.

Theorem 2. *If the throughput $r_v^{(u,v)}$ is greater than 0 for all $u, v \in V, u \neq v$, then the average throughput of Problem 1's output is higher than or equal to the average throughput achieved by any fair probabilistic assignment.*

We prove the two theorems in the Appendix.B. Note that the two theorems only holds when the throughput $r_v^{(u,v)}$ is greater than 0 for all $u, v \in V, u \neq v$. If this condition does not hold, i.e., some $r_v^{(u,v)} = 0$, then using contention to choose the followers, e.g., the method used in [4], [11], [12], would fail ZF-SIC decoding. On the other hand, to guarantee the success of ZF-SIC decoding, such a pair of clients would not be chosen in Problem 1. In addition, if we further consider the overhead of using contention to choose followers, the throughputs of the methods proposed in [4], [11], [12] would further decrease.

We next consider the 3-antenna AP scenario. Say clients u, v and w communicate with a 3-antenna AP concurrently and join the concurrent transmissions one after another. We define (u, v, w) as the *MIMO-mate* relation of clients u, v and w . The AP can use ZF-SIC to decode client w by projecting along the direction orthogonal to both clients u and v . It re-encodes client w 's stream and subtracts it from the received signals. The AP then decodes client v by projecting the resulting signal along the direction orthogonal to client u , and decodes client u after removing the signals of clients v and w . Let $r_v^{(u,v,w)}$ and $r_w^{(u,v,w)}$ denote the throughput of clients v and w , respectively. The MIMOMate matching problem in a 3-antenna AP scenario can be defined as follows:

Problem 2. (3-MIMOMate) Given a set of clients V and the throughput $r_v^{(u,v,w)}$ and $r_w^{(u,v,w)}$ for all $u, v, w \in V$, the matching problem is to find a set $M \subseteq V \times V \times V$ such that

- 1) $r_v^{(u,v,w)} > 0$ and $r_w^{(u,v,w)} > 0, \forall (u, v, w) \in M$,
- 2) $u \neq v \neq w, \forall (u, v, w) \in M$,
- 3) $u_1 \neq u_2, v_1 \neq v_2$, and $w_1 \neq w_2$ for any two distinct elements $(u_1, v_1, w_1), (u_2, v_2, w_2) \in M$,
- 4) $|M|$ is maximized,
- 5) $\sum_{(u,v,w) \in M} (r_v^{(u,v,w)} + r_w^{(u,v,w)})$ is maximized among those M satisfying Constraint 4.

Similarly, in the 3-MIMOMate problem, we are only interested in maximizing the throughput of the followers, i.e., clients v and w . We observe that the 3-MIMOMate problem, as illustrated in Figure 6(b), is actually a variation of the *Maximum 3-Dimensional Matching* problem [31], which is defined as follows: Let X, Y , and Z be disjoint sets, and let

Algorithm 1: N -MIMOMate Matching Algorithm

input: a set of clients V and the channel state information from each client to AP's N antennas

```

1 Duplicate  $V$  to  $V_1, V_2, \dots, V_N$ 
2 Remove legacy 802.11 nodes from  $V_2, \dots, V_N$ 
3 Initialize  $M \leftarrow \{\}$ 
4 for  $k := 1$  to  $N - 1$  do
5   For each edge  $(u_i, v_j) \in V_k \times V_{k+1}$ , if  $u_i$  has a predecessor
   or  $u_i \in V_1$ , set the weight of edge  $(u_i, v_j)$  to the throughput
   of  $v_j$  as it transmits concurrently with  $u_i$  and all its
   predecessors; otherwise, set the weight of  $(u_i, v_j)$  to 0
6    $M' \leftarrow$  the solution of the 2-MIMOMate matching problem
   for  $V_k \times V_{k+1}$  solved by [30]
7   if  $k = 1$  then
8     Add each  $(u_i, v_j) \in M'$  to  $M$ 
9   else
10    For each  $(u_i, v_j) \in M'$ , find the MIMO-Mate relation
    (element)  $m \in M$  that includes  $u_i \in V_k$  and add client
     $v_j \in V_{k+1}$  to the element  $m$ 
11 return  $M$ 

```

T be a subset of $X \times Y \times Z$ that includes all feasible matching combinations. The problem finds the maximum matching $M \subseteq T$ such that $u_1 \neq u_2, v_1 \neq v_2$ and $w_1 \neq w_2$ for any two distinct elements (u_1, v_1, w_1) and (u_2, v_2, w_2) in M . Hence, the differences between our 3-MIMOMate problem and the 3-dimensional matching problem are 1) our problem further considers the total weight of a matching (i.e., Constraint 5), and 2) Constraint 2 in our problem restricts each client to be included in an element at most once. For example, (u, u, v) is not a feasible combination in our problem because client u cannot transmit two streams from its single antenna at the same time.

We note that a general N -MIMOMate matching problem can be formulated in a similar manner, and it is a variation of the N -dimensional matching problem [31]. On the other hand, although the 2-MIMOMate problem is polynomial time solvable, the N -MIMOMate problem for any $N \geq 3$ is however NP-hard. We will prove the NP-hardness of the 3-MIMOMate problem in the Appendix.A by deriving a reduction from the 3-dimensional matching problem (which is also NP-hard) to our problem. The NP-hardness of the N -MIMOMate problem can be proved in a similar way.

C. Heuristic Matching Algorithm

There is an approximation algorithm [32] proposed to solve the N -dimensional matching problem. We can use the algorithm to solve our N -MIMOMate matching problem and achieve an approximation ratio, $3/2 + \epsilon$, for any $\epsilon > 0$, in terms of the size of matching. It however does not ensure to find the one achieving the maximal throughput (i.e., Constraint 5) among all maximum matchings. In addition, our problem requires an additional cost to compute the weights (throughputs) of all possible MIMO-Mates, which is an $O(|V|^N)$ computational cost. We hence propose an algorithm, as shown in Algorithm 1, to solve our MIMOMate matching problem with a reduced cost of weight computation. The basic idea of Algorithm 1 is to decompose the N -MIMOMate matching problem into $(N - 1)$ 2-MIMOMate matching problems, each

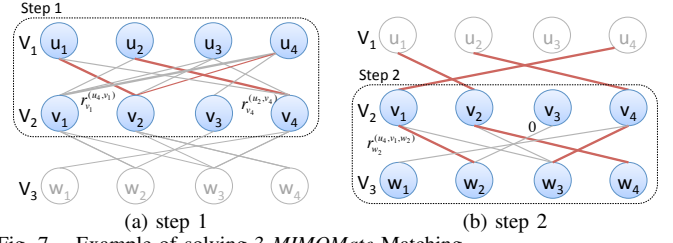


Fig. 7. Example of solving 3-MIMOMate Matching

of which can be solved by the bipartite maximum weighted maximum cardinality matching algorithm [30] in polynomial time. The advantage of such decomposition is that it reduces the cost of weight computation from $O(|V|^N)$ to $O(N|V|^2)$.

For simplicity, we use the 3-antenna AP scenario to describe our algorithm, and next explain how to generalize it to an N -antenna AP scenario. We first duplicate the client set V to V_1, V_2 and V_3 (line 1 in Algorithm 1). Our algorithm solves the 3-MIMOMate matching problem in two steps, as illustrated in Fig. 7. In the first step, as in Fig. 7(a), we set the weight of edge $(u_i, v_j) \in V_1 \times V_2$ by computing the throughput of $v_j \in V_2$ as it transmits concurrently with $u_i \in V_1$, and try to optimally match any second client in V_2 to a MIMO-Mate lead in V_1 , which is actually the 2-MIMOMate matching problem for $V_1 \times V_2$. After the first step, we determine the MIMO-Mate relation for the first and second transmitters. Then, in the second step, our goal is to add any third transmitter in V_3 into each MIMO-Mate relation. To do so, we first assign edge $(v_j, w_k) \in V_2 \times V_3$ a weight that equals the throughput of w_k when it transmits concurrently with v_j and v_j 's predecessor, which is solved in the first step. For example, in Fig. 7(b), because (u_4, v_1) is matched as MIMO-Mates in the first step, the weight of edge (v_1, w_2) equals the throughput of w_2 when it joins concurrent transmissions of u_4 and v_1 . We note that, for any $v_j \in V_2$, if it is not assigned a predecessor in the first step, we set the weight of all its outgoing edges to 0 because it is not allowed to match with any clients in V_3 in the second step. After weight assignment, we can solve another 2-MIMOMate matching problem to match clients in V_3 to clients in V_2 .

Our algorithm can be generalized to the N -MIMOMate matching problem. Specifically, it iteratively solves a 2-MIMOMate matching problem to match clients in V_{k+1} to clients in V_k , where $k = 1, 2, \dots, N - 1$. Hence, each iteration includes a client in a MIMO-Mate relation. In particular, the k^{th} iteration adds the $(k + 1)^{th}$ concurrent client to MIMO-Mates. In addition, in the k^{th} iteration, because we already know the first k clients in MIMO-Mates, we thus only need to compute the weight of edges $(u, v) \in V_k \times V_{k+1}$ according to the given MIMO-Mate relation. This is why our algorithm can reduce the cost of weight computation to $O(N|V|^2)$.

So far we describe our algorithm by assuming that all the clients are MIMOMate nodes. Our algorithm can be slightly adjusted to allow the coexistence of MIMOMate nodes and legacy 802.11 nodes. Recall that we duplicate the client set V to V_1, V_2, \dots, V_N , where N is the number of antennas equipped on the AP. Each node in the duplicated set V_i is a candidate of sending the i^{th} stream. Note that legacy nodes follow the traditional 802.11 operation and can only contend for sending the first stream. In other words, legacy nodes

do not leverage the concurrent transmission opportunities and will not follow any ongoing transmissions. We can hence simply remove those legacy nodes from V_2, \dots, V_N , and only keep them in V_1 . By doing this, legacy nodes can still use conventional 802.11 contention to occupy the first dimension, and can further be followed by some other MIMOMate nodes. Consider Fig. 7 as an example. Assume that node u_4 is a legacy node. We only put it in V_1 , but not in V_2 and V_3 . It hence can be followed by some other MIMOMate nodes, but cannot join concurrent transmissions.

V. MIMOMATE'S MEDIUM ACCESS PROTOCOL

We consider a MU-MIMO MAC protocol similar to SAM [4], where clients join the concurrent transmissions one after another. Like SAM [4], clients join concurrent transmissions one after another. Each client counts the number of concurrent streams by cross-correlating with the known preamble in the presence of ongoing transmissions. Clients can join the concurrent transmissions until they detect that the number of existing streams equals the number of antennas at the AP.² Each client determines its best bit-rate based on TurboRate, the MU-MIMO rate adaptation scheme proposed in [5]. TurboRate allows each client to announce training symbols before data transmission. All the clients who contend for the later transmissions can hence learn the channels of the ongoing streams from those training symbols, and adapt the bit-rates based on their channels. Moreover, TurboRate asks clients to give up contention opportunities if their SNR after ZF-SIC decoding is too low to be decodable. To increase the gain of MU-MIMO, the protocol forces concurrent clients to end their transmissions at about the same time. To do so, concurrent clients overhear the information about the frame duration of the first stream, which is embedded in the MAC header, and fragment or aggregate their packets accordingly [5] [23]. MIMOMate differs from the existing MAC protocols in that it only allows clients to use 802.11's CSMA/CA to contend for the first stream, but lets the remaining clients join the concurrent transmission of its predecessor in the *MIMO-Mate* relation scheduled by Algorithm 1. In particular, say a client is scheduled to transmit the k^{th} stream in the *MIMO-Mates*; it can start transmitting once it detects $k - 1$ preambles from all its predecessors after its leader wins the contention. Hence, all clients in the *MIMO-Mates* only require one contending process.

To realize such a user matching protocol, MIMOMate's MAC needs two major modifications: 1) the AP needs to learn the uplink channel information of all its clients, and 2) the AP needs to announce the matching result to its clients. To learn the channel information, one possible solution is to let all the clients learn their uplink channels and report this information to the AP. To do so, the clients leverage *channel reciprocity* [34], which refers to the property that the channels in the forward and reverse directions are the same. Using reciprocity, every client can exploit the beacons to learn the

²The preamble-counting based protocol, like SAM [4], could suffer from collisions when hidden nodes interrupt the preamble-counting process. We apply a multi-round light-weight handshaking mechanism proposed in [33] to address the hidden terminal problem with minimum overhead.

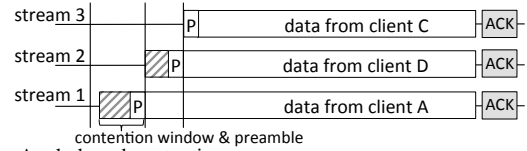


Fig. 8. Angle-based contention

downlink channel and use it to estimate the uplink channel. It is however an expensive overhead to ask all the clients to report their channels for every packet transmission. On the other hand, legacy nodes, which follow the traditional 802.11 operation, do not feedback this information. We hence perform the following optimizations to reduce the overhead of channel feedback: The AP learns the uplink channel of a client from its uplink frames, including the association frames when that client joins the network, the data frame of its uplink packets, and the ACK of its downlink packets. The AP hence only needs to re-schedule *MIMO-Mates* when it detects that the channels of certain clients change due to channel variation or user mobility. Once the AP reschedules *MIMO-Mates*, it announces the updated matching result to the MIMOMate nodes. A simple solution is to annotate the periodical beacon messages with the announcement. However, legacy nodes might not be able to identify the modified beacon format. To enable the coexistence of MIMOMate nodes and legacy nodes, we let the AP send the matching result in another control frame using the subtype not used in conventional 802.11 [35].

We notice that the above protocol only operates properly if the scheduled *MIMO-Mates* always have traffic to transmit. However, in practice, a client might have a bursty traffic pattern. Hence, to enable users to fully utilize concurrent transmission opportunities, we can further integrate MIMOMate with any contention-based MAC protocol. Specifically, clients can contend for the unused degrees of freedom if the scheduled *MIMO-Mates* do not have traffic to transmit. However, to better exploit the gain of concurrent transmissions, we propose to integrate MIMOMate with an *angle-based* contention scheme. In particular, when any of the scheduled *MIMO-Mates* does not have traffic to transmit, we allow other clients to contend for the concurrent transmission opportunity, e.g., the second stream in the example shown in Fig. 8, until the number of concurrent streams equals the number of antennas supported by the AP, N . However, we modify the contention mechanism to assign different users a different probability of winning a concurrent transmission opportunity, according to their channel orthogonality with the ongoing streams. Specifically, we tend to let a client with a larger angle between its channel and the channels of the ongoing streams have a higher probability to win the concurrent transmission opportunity such that SNR reduction due to projection can be minimized.

To achieve this goal, we apply Algorithm 2 to adjust the contention window for each concurrent stream according to the channels of the concurrent transmitters. Specifically, when a node contends for sending the k^{th} stream in the presence of $(k - 1)$ ongoing transmissions, it will adjust its contention windows based on the angle between its own channel and the channels of the $(k - 1)$ ongoing transmitting clients. We assume that clients can learn the angle between its own

Algorithm 2: Angle-based Contention Scheme

input: the initial contention window of the k^{th} stream
 $CW^k \leftarrow CW_{min}$; the initial update of the k^{th} stream
 $\delta_{cur}^k \leftarrow 0$; N antennas at the AP

```
1 for  $1 \leq k \leq N$  do
2   //contention for the  $k^{th}$  stream in each packet transmission
3   Learn the angle  $\theta \in [0, \pi/2]$  between the client's channel
   and the channels of the  $(k-1)$  ongoing streams
4   if  $SNR \text{ after projection} \leq 802.11 \text{ SNR regime}$  then
5     return; //give up this contention
6   Update  $CW^k$  using traditional 802.11 backoff
7   if  $k > 1$  then
8      $\delta_{last}^k \leftarrow \delta_{cur}^k$ 
9      $CW_{orig}^k \leftarrow CW^k - \delta_{last}^k$ 
10     $CW^k \leftarrow CW_{orig}^k - \frac{\theta - \pi/4}{\pi/4} * CW_{orig}^k$ 
11     $CW^k \leftarrow \max(CW_{min}, \min(CW_{max}, CW^k))$ 
12     $\delta_{cur}^k \leftarrow CW^k - CW_{orig}^k$ 
```

channel and the channels of the ongoing transmitters using the distributed method proposed in [5]. The high-level idea of the angle-based contention is that, if this angle is large, we let the client decrease its contention window and hence earn a higher probability to win the concurrent transmission opportunity. Otherwise, the client gives other clients a higher priority to transmit concurrently by increasing its contention window.

To realize the above design, we let each client maintain a distinct contention window CW^k for the contention of the k^{th} stream. The contention windows are adjusted according to the channels of the ongoing clients. The amount of increment (or decrement) is proportional to the inter-client angle, i.e. $\frac{\theta - \pi/4}{\pi/4}$ in line 10. To ensure fairness, we ask a client assigned a higher priority in the current packet to pay back its opportunity in the next packet. To this end, if a client decreases (increases) the contention window by δ for the current packet, it pays (earns) the priority back by increasing (decreasing) δ to its contention window for the next packet, i.e., δ_{last} in line 8. The above contention scheme can be applied for the contention of each concurrent stream until the number of streams reaches the number of antennas supported by the AP, i.e., $1 \leq k \leq N$.

Overhead and complexity: Recall that implementing MIMOMate as a MAC protocol relies on two modifications: 1) the AP needs to learn the uplink channels, and 2) the AP needs to announce the matching results. Note that we let the AP measure the channel information from historical uplink frames without any additional message overhead. The only additional overhead required by our design is the matching announcement. We will show in Section VI that such a small overhead does not offset the gain of our matching algorithm. On the other hand, since our matching algorithm is performed in the access point, and the complexity of clients should not change much. Therefore, the only supports we need from MIMOMate clients are that 1) they need to receive the matching announcement, and 2) they need to adapt the contention window size based on a simple operation defined in our angle-based contention scheme, as in Algorithm 2. We believe additional power consumption in the clients due to our design should be negligible.

VI. EXPERIMENTAL RESULTS

We build a prototype of MIMOMate using the USRP-N200 radio platform, which is equipped with an RFX2400 daughterboard. A multi-antenna AP is built by combining multiple USRP-N200 boards using an external clock. We implement an OFDM PHY layer with standard 802.11 modulations (BPSK, 4-64QAM) and code rates. Since USRP-N200 operates on a 10MHz channel, the possible bit rates range from 3 to 27 Mb/s. We evaluate the performance of MIMOMate in both 2-antenna and 3-antenna AP scenarios. Limited by the number of USRPs we have, we set 6 clients to contend for transmitting two packets concurrently to the 2-antenna AP, while setting 5 clients to contend for transmitting three packets concurrently to the 3-antenna AP. To allow multiple clients to transmit concurrently, we leverage the synchronization method used in [5] [23]. Specifically, for each experiment, the AP broadcasts a trigger signal. Each client records the timestamp of detecting the trigger, $t_{trigger}$, waits a pre-defined period of time, t_{Δ} , and sets the timestamp of the beginning of its transmission to $t_{start} = t_{trigger} + t_{\Delta}$. In our testbed, t_{Δ} is set to 0.1s, which is long enough to tackle the delays introduced by software.

We compare the following schemes: 1) MIMOMate, which is our proposed protocol, 2) *max-throughput first*, which always allows the client that achieves the maximal throughput after projection to join the concurrent transmissions, 3) *max-angle first*, which always allows the client that has the maximum angle with the ongoing transmissions to transmit concurrently, 4) SAM [4], i.e., *contention-based* protocol without RTS/CTS, which assigns all users an equal probability to sequentially contend for each concurrent transmission opportunity, and 5) MRC [11], i.e., multi-round contention, which also assigns each client an equal probability of winning contentions, but precedes concurrent transmissions with multiple rounds of RTS and a single CTS. For all the comparison schemes, we apply TurboRate [5], a MU-MIMO rate adaptation scheme, to allow concurrent clients to select their best bit rates.

Due to the timing constraints limited by software radio, we do not implement contention, random backoff and ACK in USRPs. Instead, for each experiment, we offline create a packet trace of 1,000 1500-byte packets for each client. The traces of different clients are generated based on the above four comparison schemes, and ensure that there are at most 2 and 3 clients assigned to transmit concurrently in a particular time-slot in 2- and 3- antenna AP scenarios, respectively. In particular, in the beginning of each experiment, we let each client transmit training symbols, one after another, for the AP to estimate its uplink channel. The AP then performs offline contention to generate 1,000 rounds of concurrent transmissions. For all the comparison schemes, in each round of concurrent transmissions, the AP assigns each client a randomly-selected backoff value between 1 and its contention window, and picks the client with the smallest backoff value to send the first stream. The contention window of each client is updated according to the 802.11 standard if collisions occur. The AP then assigns the remaining concurrent transmission

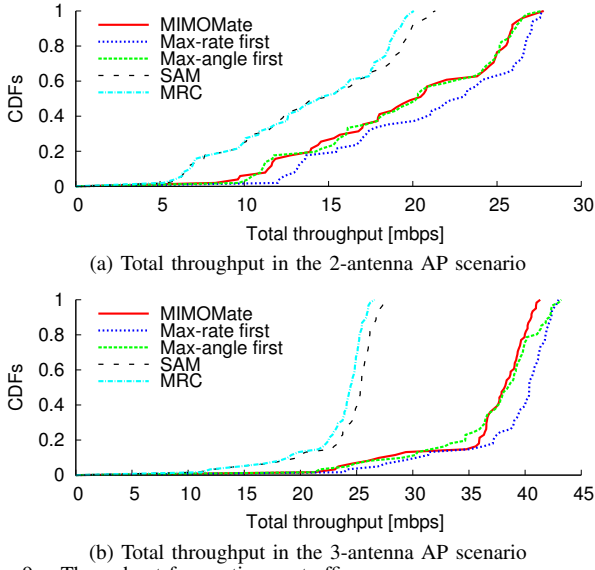


Fig. 9. Throughput for continuous traffic

opportunities to other clients based on design of various comparison protocols. For example, in SAM, the AP uses the same contention scheme to assign the remaining transmission opportunities; in MIMOMate the AP assigns *MIMO-Mates* of the first contention winner to transmit concurrently; in *max-throughput first* and *max-angle first*, the AP assigns the remaining transmission opportunities based on the throughput and the angle between channels, respectively. Based on the contention results, the AP generates the packet trace for each client, and immediately sends the trace to each client through Ethernet connection. Each USRP client can hence read its offline-generated packet trace and determine the time that it should transmit packets accordingly. Clients are asked to send null symbols, i.e., 0, if they are not selected to transmit in a round of packet transmissions. Since offline contention performed in the AP does not take too much time, we expect that the channels do not change significantly, i.e., the channels during data transmissions would be similar to those learned in the training phase. In addition, since USRPs cannot implement real-time ACK, we disable retransmissions in the experiments. That is, the AP simply drops a packet if the packet cannot be received or decoded correctly.

We first evaluate the performance of MIMOMate when clients have a continuous traffic pattern, and next evaluate the performance of integrating the angle-based contention mechanism (Algorithm 2) with MIMOMate when clients have a bursty traffic pattern.

A. Performance Comparison for Continuous Traffic

We evaluate the performance of the comparison schemes in terms of 1) throughput gain, 2) fairness, and 3) overhead.

Throughput gain: We first check the throughput gain delivered by MIMOMate when users have a continuous traffic pattern, i.e., always have packets to send. Hence, in MIMOMate, the scheduled *MIMO-Mates* can always transmit concurrently if their lead wins the first contention. We repeat the experiment with random assignment of client locations in our testbed.

Figs. 9(a) and 9(b) plot the CDFs of the total throughput in

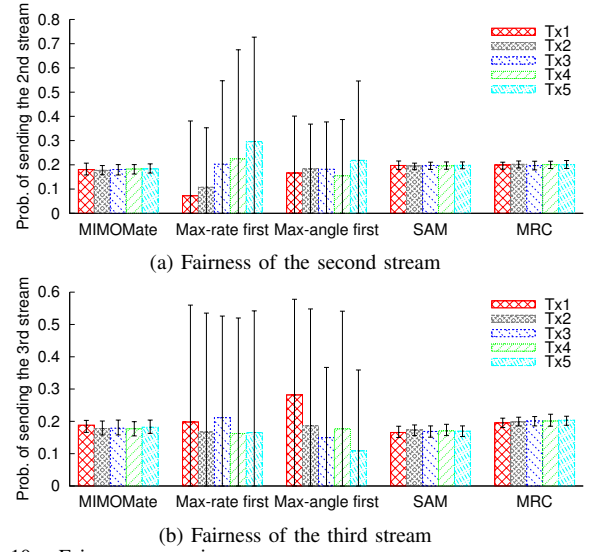


Fig. 10. Fairness comparison

2-antenna and 3-antenna scenarios, respectively. The figures show that traditional contention-based protocols, i.e., SAM and MRC, assign each user an equal probability to win the contention, without considering the channel characteristics, and produce a low throughput. MRC requires additional RTS-CTS overhead, and hence performs a little bit worse than SAM. Compared to SAM (MRC), the average throughput gain from enabling concurrent transmissions with MIMOMate's user selection is about 42% (45%) and 52% (57%) in 2- and 3-antenna AP scenarios, respectively. The gain mainly comes from two design principles in MIMOMate: 1) minimizing SNR reduction due to MIMO decoding, and 2) reducing the channel time wasted for contending for concurrent transmissions. Note that the gain in the 3-antenna AP scenario is higher than that in the 2-antenna AP scenario. It implies that user matching plays an important role to deliver the MU-MIMO gain especially when the number of concurrent transmissions supported by the system increases. The figures also show that *max-angle first* and *max-throughput first* produce a throughput comparable to (or even slightly higher than) our MIMOMate because they greedily select the users with the best channel characteristics or with the highest throughput to join the concurrent transmissions. In addition, similar to MIMOMate, they also require only one contending process. We will show later that these two schemes however result in unfair resource sharing.

Fairness: We next examine fairness of sharing concurrent transmission opportunities among clients in a 3-antenna AP scenario. We plot in Figs. 10(a) and 10(b) the number of the second transmission opportunities and the third transmission opportunities obtained by each client over the total number of transmissions, which is the metric used to evaluate fairness in our experiments. The figures show that both the contention-based schemes, i.e., SAM and MRC, and our MIMOMate enable all clients to get almost an equal probability to transmit the second stream and the third stream, respectively. This implies that our matching algorithm enables users to achieve the same level of fairness as if they use a fair contention mechanism. The probability of sending the third stream in

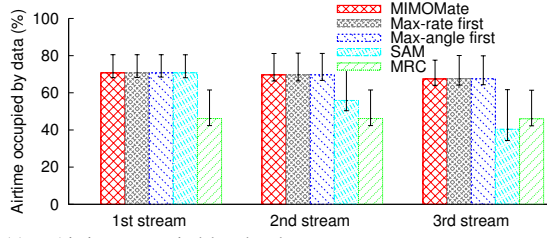


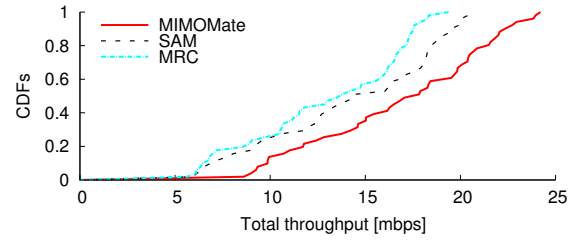
Fig. 11. Airtime occupied by the data streams

SAM is however slightly lower than that in MIMOMate and MRC. This is because, if the transmission time of the first stream is too short due to a high data rate, then there might be no enough time for SAM to hold the third stream and its contention. On the other hand, in *max-throughput first* and *max-angle first*, users cannot have a fair opportunity to transmit concurrently because these two schemes always favor certain users to achieve a high throughput. Based on the results in Figs. 9 and 10, we conclude that MIMOMate achieves a throughput comparable to the greedy algorithms, while providing users a fairness level similar to the contention mechanism.

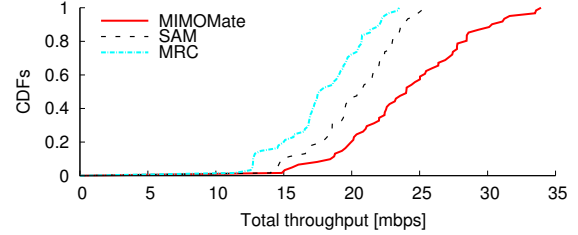
Overhead: We now compare the overhead of different systems. Recall that we do not implement ACK and contention in USRPs. Thus, we offline compute the channel time occupied by the protocol overhead. To do so, we feed the throughput outputted by the experiments (i.e., the rate of the data frames without considering the 802.11 overhead) to the offline computation, and add the overhead of each protocol, including contention, interframe timing (SIFS/DIFS), the PLCP/MAC headers, and the ACK, to each packet transmission. Fig. 11 plots the percentage of airtime occupied by the data frames, which is computed by the ratio of airtime for data transmission to the overall airtime occupied by a packet (i.e., including the overhead). The figure shows that, since clients in our MIMOMate and SAM use 802.11's contention to compete for sending the first stream, their overhead for the first stream is the same with that of the conventional 802.11. By eliminating the contending process for the second stream, MIMOMate and the greedy algorithms can utilize about 63% of airtime to transmit the second streams. In contrast, both SAM and MRC require multiple rounds of contention, which significantly offset the MIMO gain when the number of antennas supported by the AP keeps increasing. SAM allows each client to transmit immediately once it wins the contention. Hence, the airtime of data sent in a higher dimension becomes shorter and shorter. For MRC, all concurrent clients start their transmissions after receiving the CTS, their available airtime is hence shorter yet the same. However, since multiple clients might send RTS concurrently in MRC, the number of contention rounds required in MRC might be fewer than that in SAM. This explains why the airtime of the third stream in MRC is longer than that in SAM.

B. Throughput Gain for Bursty Traffic

We next evaluate the performance of MIMOMate when clients have a bursty traffic pattern. The packet traces are generated using the following model. Each user transmits several files to the AP, and the size of each file is randomly



(a) Total throughput in the 2-antenna AP scenario



(b) Total throughput in the 3-antenna AP scenario

Fig. 12. Throughput for bursty traffic

selected from 500 to 550 KB. The arrival of file transmission follows a Poisson process with an arrival rate $\lambda = 2$ files per second. Thus, when any of the scheduled *MIMO-Mates* does not have traffic to send, other clients contend for transmitting concurrently using the contention window computed based on Algorithm 2. Fig. 12 plots the CDFs of the total throughput. The figure shows that, compared to SAM (MRC), the average throughput gain achieved by combining MIMOMate with angle-based contention is about 22% (33%) and 19% (36%) for 2- and 3-antenna scenarios, respectively. The throughput gain in this case is lower than that in the continuous traffic scenario because angle-based contention introduces additional contention overhead. Also, for some periods, there are only a few clients with traffic demand and hence concurrent transmission opportunities cannot be fully utilized. The gap between SAM and MRC for bursty traffic is larger than that for continuous traffic. This is because, in MRC, when there are always at least N contending clients, where N is the degrees of freedom, the AP might be able to detect N clients within fewer than N contention rounds, if some clients send RTS at the same time in one contention round. However, when the number of contending clients is less than N , the AP always needs to wait for the duration of N rounds of RTS and then responds the CTS.

VII. SIMULATION RESULTS

We further perform simulations to evaluate the performance of MIMOMate in large-scale scenarios. The simulations are designed to answer the following questions.

- How does MIMOMate perform in different scales of networks?
- How does the packet size affect the throughput performance of comparison protocols?
- Can legacy 802.11 devices operate normally in the presence of MIMOMate nodes?

In each simulation, we uniformly randomly distribute the users in a disk with center the AP and radius 100 m. Furthermore, the antennas on the AP are collinear with a gap of 0.05 m between two neighboring antennas. The channels are

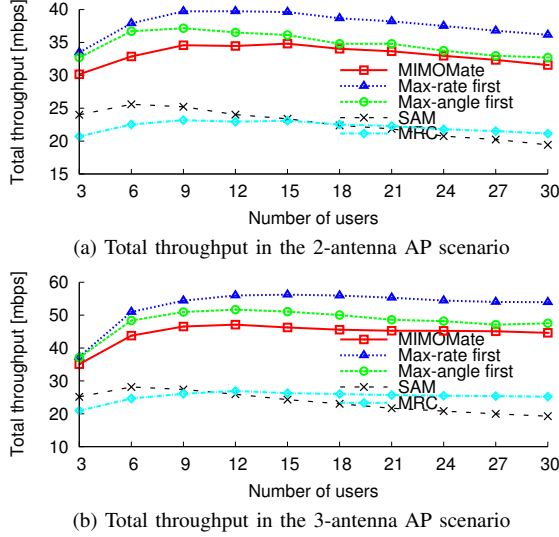


Fig. 13. Impact of number of users on throughput

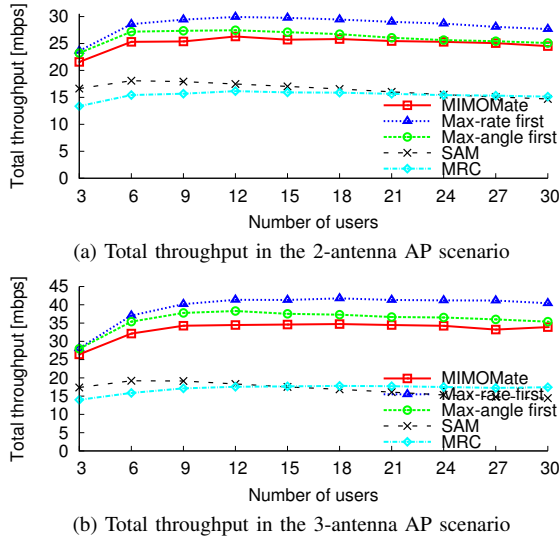


Fig. 14. Impact of various packet sizes on throughput

generated according to the Rayleigh fading channel model, and we assume the available transmission bit-rates are 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s, which are identical with 802.11a. The default number of users is set to 15, and the default packet size is set to 1500 bytes. The detailed settings will be specified in each simulation.

A. Impact of Number of Clients

We evaluate the performance of MIMOMate when the number of clients varies from 3 to 30. Figs. 13(a) and 13(b) plot the total throughput for 2-antenna and 3-antenna AP scenarios, respectively. The trend of the simulation results are similar to that of small-scale experimental results. The effect of increasing the number of clients on MIMOMate, *max-angle first*, and *max-throughput first* is relatively small, showing that MIMOMate operates well even when the network scales up. One thing worth noting is that SAM outperforms MRC when the number of clients is small, e.g., less than twelve, because its clients do not need to wait for transmitting concurrently

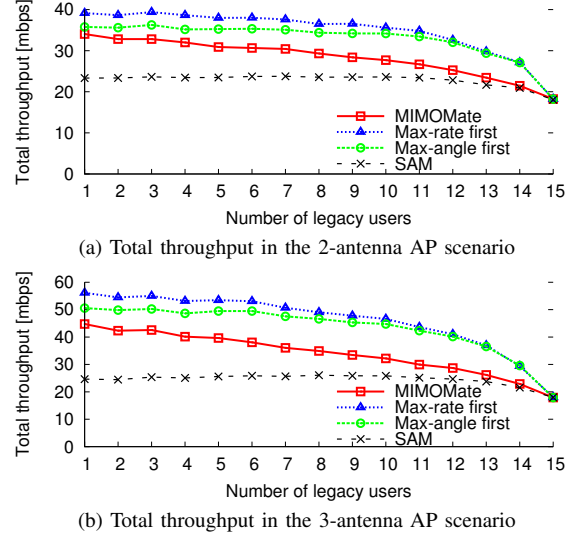


Fig. 15. Impact of number of legacy nodes on throughput

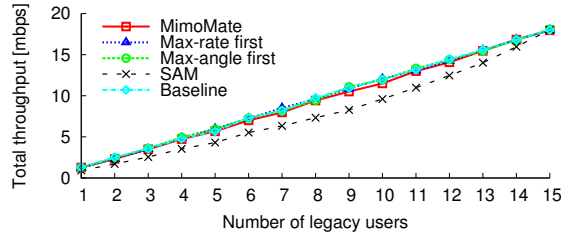
after multiple rounds of contention. The throughput of SAM however decreases when the number of clients increases. The reason is that, although both SAM and MRC require multiple rounds of contention, contention failure can only occur in the last round of contention in MRC, yet could happen in any round of contention in SAM. The more clients exist, the higher probability that contention fails is. When contention fails, the number of concurrent streams would exceed the degrees of freedom, which makes ZF-SIC decoding fail. Hence the result.

B. Impact of Dynamic Packet Sizes

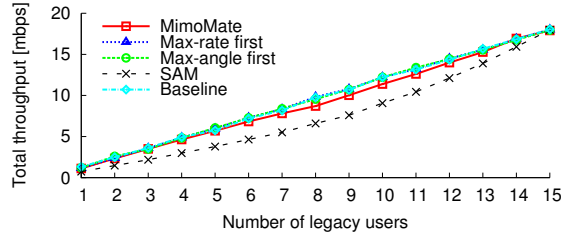
We next evaluate how MIMOMate performs when the packet size varies dynamically. In this simulation, we uniformly randomly pick a size between 200 bytes and 1500 bytes for the first client of each transmission. The clients joining later end their transmissions at the same time with the first stream. Again, Figs. 14(a) and 14(b) plot the total throughput for 2-antenna and 3-antenna AP scenarios, respectively. Due to a smaller average packet size, i.e., around $(200 + 1500)/2$ bytes, and, as a result, a higher proportion of airtime occupied by the overhead, the throughput in this simulation is less than those found in Fig. 13; However, the advantage of MIMOMate does not get affected when the packet size changes. Observer that, compared to Figs. 13(a) and 13(b), the throughput of SAM decreases slower here. This is because, in SAM, when the packet size is small, then there might be no remaining airtime for clients to exploit the transmission opportunities of high dimensions. In other words, the number of contentions for SAM in this simulation is less than that in the previous simulation. As a result, less contention failures occur here. Therefore, compared with the previous simulation, the effect of increasing the number of users on SAM is less here.

C. Impact of Existence of Legacy Devices

We finally check the performance of MIMOMate in the presence of legacy nodes. Since MRC is not compatible with the traditional 802.11 standard, we exclude it from this simulation. In this simulation, we set the total number of



(a) Throughput of legacy nodes in the 2-antenna AP scenario



(b) Throughput of legacy nodes in the 3-antenna AP scenario

Fig. 16. Average throughput of legacy devices

clients at 15, and let some of nodes be legacy 802.11 clients. Figs. 15(a) and 15(b) show the total throughput when the number of legacy nodes varies from 1 to 15. The results show that, in MIMOMate *max-angle first* and *max-throughput first*, the total throughput decreases as the number of legacy nodes increases. The first reason is that, since legacy nodes can only send the first stream, i.e., occupying the first dimension, the probability of picking concurrent clients with a good channel decreases when the number of non-legacy nodes decreases. Second, when the number of non-legacy clients is less than the degrees of freedom, the concurrent transmission opportunities might not be able to be fully utilized, as a result reducing the throughput significantly. When all the users are legacy users, i.e., the number of legacy users is fifteen, all the methods degenerate to the traditional 802.11 protocol and hence perform the same. The performance of SAM however does not change much with various numbers of legacy users because it simply randomly picks clients to fully utilize the available degrees of freedom, without considering channel orthogonality between concurrent clients. On the contrary, the performance of SAM increases slightly when the number of non-legacy nodes decreases, because the probability of collisions due to contention failure decreases when more nodes join contention.

We further check whether the throughput performance of legacy nodes gets affected by our matching design. Figs. 16(a) and 16(b) compare the total throughput of legacy nodes in comparison schemes with that in the traditional 802.11 protocol. The results show that their throughput in SAM is significantly worse than that in traditional 802.11. This is because SAM suffers from contention failures that might happen in the second and third streams. The figure also shows that legacy nodes can coexist with MIMOMate nodes well and achieve a similar performance, as compared to that in traditional 802.11. A small gap between MIMOMate and traditional 802.11 is due to occasional contention failures that might happen when contention is required for some first winners who are not assigned any follower.

VIII. CONCLUSION

This paper introduces MIMOMate, a user matching protocol that maximally delivers the gain of a MU-MIMO LAN, while, at the same time, ensuring all the clients to have a fair opportunity to transmit concurrent packets. The clients scheduled as the *MIMO-Mates* can join concurrent transmissions one after another with only one contending process, as a result reducing the MAC overhead significantly. We also integrate MIMOMate with an angle-based contention mechanism to best utilize concurrent transmission opportunities when any of the scheduled *MIMO-Mates* does not have traffic to transmit. Our prototype implementation shows that MIMOMate increases the throughput by 42% and 52% over the contention-based protocol for 2- and 3-antenna AP scenarios, respectively, and also provides fairness for clients. Our theoretical analysis also proved that MIMOMate can always achieve a higher or equal throughput as compared to the traditional contention scheme in the 2-antenna AP scenario. Analytic performance evaluation for any general scenarios is considered as our future work.

REFERENCES

- [1] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly, "Design and Experimental Evaluation of Multi-User Beamforming in Wireless LANs," in *ACM MobiCom*, 2010.
- [2] H. Yu, L. Zhong, A. Sabharwal, and D. Kao, "Beamforming on Mobile Devices: a First Study," in *ACM MobiCom*, 2011.
- [3] H. Rahul, S. Kumar, and D. Katabi, "MegaMIMO: Scaling Wireless Capacity with User Demands," in *ACM SIGCOMM*, 2012.
- [4] K. Tan, H. Liu, J. Fang, W. Wang, J. Zhang, M. Chen, and G. M. Voelker, "SAM: Enabling Practical Spatial Multiple Access in Wireless LAN," in *ACM MobiCom*, 2009.
- [5] W.-L. Shen, Y.-C. Tung, K.-C. Lee, K. C.-J. Lin, S. Gollakota, D. Katabi, and M.-S. Chen, "Rate Adaptation for 802.11 Multiuser MIMO Networks," in *ACM MobiCom*, 2012.
- [6] H. S. Rahul, S. Kumar, and D. Katabi, "JMB: scaling wireless capacity with user demands," in *ACM SIGCOMM*, 2012.
- [7] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong, "Argos: practical many-antenna base stations," in *ACM MobiCom*, 2012.
- [8] D. Tse and P. Vishwanath, *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.
- [9] Y. Zhang, C. Ji, Y. Liu, W. Malik, D. O'Brien, and D. Edwards, "A Low Complexity User Scheduling Algorithm for Uplink Multiuser MIMO Systems," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2486–2491, July 2008.
- [10] Y. Kim, S. Cho, and D. K. Kim, "Low Complexity Antenna Selection based MIMO Scheduling Algorithms for Uplink Multiuser MIMO/FDD System," in *IEEE VTC Spring*, 2007.
- [11] Y. Zhang, "Multi-Round Contention in Wireless LANs with Multipacket Reception," *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, pp. 1503–1513, 2010.
- [12] P. X. Zheng, Y. Zhang, and S.-C. Liew, "Multipacket Reception in Wireless Local Area Networks," in *IEEE International Conference on Communications (ICC)*, 2006.
- [13] D. Jung, R. Kim, and H. Lim, "Asynchronous Medium Access Protocol for Multi-User MIMO Based Uplink WLANs," *IEEE Transactions on Communications*, vol. 60, no. 12, pp. 3745–3754, 2012.
- [14] 3GPP TSG RAN1 #46, R1-062074, "Link Simulation Results for Uplink Virtual MIMO," Tallinn, Estonia, Aug. 2006.
- [15] 3GPP TSG-RAN1 #46, R1-062052, "UL system analysis with SDMA," Tallinn, Estonia, Aug. 2006.
- [16] Y. Chao, F. Bin, Q. Yu, and W. Wen-bo, "Adjustable Determinant Pairing Scheduling for Virtual MIMO System," in *Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007. International Symposium on*, 2007.
- [17] X. Chen, H. Hu, H. Wang, H.-H. Chen, and M. Guizani, "Double Proportional Fair User Pairing Algorithm for Uplink Virtual MIMO Systems," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, 2008.

- [18] "Ettus Inc., Universal Software Radio Peripheral," <http://ettus.com>.
- [19] A. El Gamal and T. Cover, "Multiple User Information Theory," *Proceedings of the IEEE*, vol. 68, no. 12, pp. 1466 – 1483, dec. 1980.
- [20] D. Gesbert, M. Kountouris, R. Heath, C.-B. Chae, and T. Salzer, "Shifting the MIMO Paradigm," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 36 – 46, sept. 2007.
- [21] P. Viswanath and D. Tse, "Sum Capacity of The Vector Gaussian Broadcast Channel and Uplink-downlink Duality," *IEEE Transactions on Information Theory*, vol. 49, no. 8, pp. 1912 – 1921, aug. 2003.
- [22] S. Gollakota, S. D. Perli, and D. Katabi, "Interference Alignment and Cancellation," in *ACM SIGCOMM*, 2009.
- [23] K. C.-J. Lin, S. Gollakota, and D. Katabi, "Random Access Heterogeneous MIMO Networks," in *ACM SIGCOMM*, 2011.
- [24] H. Senhwa, Q. Ling, and S. Lin, "Zero-Forcing Beamforming with Receiver Antenna Selection in Downlink Multi-Antenna Multi-User System," *Journal of Systems Engineering and Electronics*, vol. 19, no. 6, pp. 1258 – 1263, dec. 2008.
- [25] A. Liu, W. Luo, and X. Haige, "Efficient User Selection and Generalized Beamforming for Multi-User MIMO Downlink," in *IEEE VTC-Fall*, 2008.
- [26] H. Viswanathan, S. Venkatesan, and H. Huang, "Downlink Capacity Evaluation of Cellular Networks with Known-interference Cancellation," *Selected Areas in Communications, IEEE Journal on*, vol. 21, no. 5, pp. 802–811, 2003.
- [27] C. Guthy, W. Utschick, and G. Dietl, "A User Grouping Method for Maximum Weighted Sum Capacity Gain," in *IEEE ICC*, 2009.
- [28] M. Sharif and B. Hassibi, "On the Capacity of MIMO Broadcast Channels with Partial Side Information," *IEEE Transactions on Information Theory*, vol. 51, no. 2, pp. 506–522, 2005.
- [29] O. Aboul-Magd, U. Kwon, Y. Kim, and C. Zhu, "Managing Downlink Multi-User MIMO Transmission Using Group Membership," in *IEEE Consumer Communications and Networking Conference (CCNC)*, 2013.
- [30] K. Mehlhorn and S. Näher, *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999. [Online]. Available: <http://www.mpi-sb.mpg.de/~mehlhorn/LEDAbook.html>
- [31] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [32] C. Hurkens and A. Schrijver, "On The Size of Systems of Sets Every t of Which Have An SDR, With An Application to The Worst-Case Ratio of Heuristics for Packing Problems," *SIAM Journal on Discrete Mathematics*, vol. 2, no. 1, pp. 68–72, 1989.
- [33] K. Lin, Y.-J. Chuang, and D. Katabi, "A light-weight wireless handshake," *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 2, pp. 28–34, Mar. 2012.
- [34] M. Guillaud, D. Slock, and R. Knopp, "A Practical Method for Wireless Channel Reciprocity Exploitation through Relative Calibration," in *Signal Processing and Its Applications*, 2005.
- [35] M. S. Gast, *802.11 Wireless Networks*, 2nd ed. O'Reilly, 2005.
- [36] L. Lovász and M. Plummer, *Matching Theory*. North-Holland, 1986.

APPENDIX

A. Proof of the NP-hardness of the 3-MIMOMate Problem

Theorem 3. *The 3-MIMOMate problem is NP-hard.*

Proof: We start our proof by relaxing the 3-MIMOMate problem to a simpler problem, denoted by the maximum fairness (3MF) problem, which finds a set M such that Constraints 1–4 in problem 2 are satisfied. Any optimal solution of the 3MF problem is hence a feasible solution of our problem. We then proceed the proof by showing that even the decision version of the relaxed 3MF problem is NP-hard.

Let $3MF(V, r, k)$ denote an instance of the decision version of the 3MF problem, which seeks a matching $M \subseteq V \times V$ with $|M| \geq k$. We prove its NP-hardness by polynomial-time reduction from the 3-dimensional matching problem, which is NP-hard. Let $3DM(X, Y, Z, T, k)$ denote the decision version of the 3-dimensional matching problem, which finds a matching $M \subseteq T$, where $T \subseteq X \times Y \times Z$, such that $u_1 \neq u_2, v_1 \neq v_2$, and $w_1 \neq w_2$ for any $(u_1, v_1, w_1), (u_2, v_2, w_2) \in M$ and $|M| \geq$

k . For every instance $3DM(X, Y, Z, T, k)$, we can construct an instance $3MF(V, r, k)$ in polynomial time as follows:

- 1) $V = \{u_i | 1 \leq i \leq |X| + |Y| + |Z|\}$.
- 2) Set $r_{u_j}^{(u_i, u_j, u_k)} + r_{u_k}^{(u_i, u_j, u_k)} = 1, \forall (u_i, u_j, u_k), \text{ if } (x_i, y_j - |X|, z_k - |X| - |Y|) \in T$; otherwise, set it to 0.

We show that $3DM(X, Y, Z, T, k)$ has a feasible solution M if, and only if, $3MF(V, r, k)$ has a feasible solution M' . For the "only if" direction, for all the elements (x_i, y_j, z_k) in M , we add $(u_i, u_{j+|X|}, u_{k+|X|+|Y|})$ to the solution of $3MF(V, r, k)$, M' . This solution is feasible because, by the construction of the instance, $r_{u_{j+|X|}}^{(u_i, u_{j+|X|}, u_{k+|X|+|Y|})} + r_{u_{k+|X|+|Y|}}^{(u_i, u_{j+|X|}, u_{k+|X|+|Y|})} = 1$, and thus Constraints 1 and 3 of the 3MF problem hold. Constraint 2 is also satisfied, i.e., $u_i, u_{j+|X|}$ and $u_{k+|X|+|Y|}$ are distinct clients, because $i \leq |X| < j+|X| \leq |X|+|Y| < k+|X|+|Y|$. Finally, since different elements (x_i, y_j, z_k) map to different $(u_i, u_{j+|X|}, u_{k+|X|+|Y|})$, then $|M'| = |M| \geq k$.

For the "if" direction, for every $(u_i, u_j, u_k) \in M'$, we add $(x_i, y_j - |X|, z_k - |X| - |Y|)$ to the solution of $3MF, M$. Since (u_i, u_j, u_k) in M' , we have $r_{u_j}^{(u_i, u_j, u_k)} + r_{u_k}^{(u_i, u_j, u_k)} = 1$. Thus, by the construction of the instance, $(x_i, y_j - |X|, z_k - |X| - |Y|) \in T$. In addition, by Constraint 3, the constraint of the 3DM problem that restricts $u_1 \neq u_2, v_1 \neq v_2$, and $w_1 \neq w_2$, for any two distinct elements $(u_1, v_1, w_1), (u_2, v_2, w_2) \in T$, holds. Finally, again, since $|M| = |M'|$, $M \geq k$ holds as well. Hence, we conclude that the 3FM problem is also NP-hard. ■

B. Proofs of Theorem 1 and Theorem 2

Since Theorem 2 implies Theorem 1, we only need to show Theorem 2. Furthermore, it is sufficient to show that the average throughput of the output of Problem 1, denoted as T_M , is greater than or equal to that of the optimal fair probabilistic assignment selection (the fair probabilistic assignment selection that achieves the highest average throughput), denoted as T_R . For ease of presentation, we give an order to the clients, so that the input V in Problem 1, i.e., the set of all clients, is $\{1, 2, \dots, |V|\}$. Note that since the throughput $r_j^{(i,j)}$ is greater than 0 for all $i, j \in V, i \neq j$, i.e., ZF-SIC decoding is successful, a client who transmits the first stream can get about the same throughput no matter who its follower is [5]. In addition, every client has an equal probability of winning the first contention. Therefore, we assume that the throughputs contributed by the first stream are the same in both the output of Problem 1 and the optimal fair probabilistic assignment. Thus, we can ignore the average throughput contributed by the first stream in the following proof, since we only need to show $T_M \geq T_R$.

To derive the average throughput, we introduce a variable $p^{i,j}$ for all $i, j \in V, i \neq j$. $p^{i,j}$ represents the probability that client j is chosen as client i 's follower. Therefore, given $p^{i,j}$ s, the average throughput can be expressed as follows.

$$\begin{aligned} \sum_{i \in V} \{Pr\{\text{client } i \text{ wins the first contention}\} \sum_{j \in V \setminus \{i\}} p^{i,j} r_j^{(i,j)}\} \\ = \frac{1}{|V|} \sum_{i \in V} \sum_{j \in V \setminus \{i\}} p^{i,j} r_j^{(i,j)}. \end{aligned}$$

Now, we are ready to derive T_M and T_R . Denote $p_M^{i,j}$ s and $p_R^{i,j}$ s as the $p^{i,j}$ s used in the output of Problem 1 and the optimal fair probabilistic assignment, respectively. It is then sufficient to show that

$$\sum_{i \in V} \sum_{j \in V \setminus \{i\}} p_M^{i,j} r_j^{(i,j)} \geq \sum_{i \in V} \sum_{j \in V \setminus \{i\}} p_R^{i,j} r_j^{(i,j)}. \quad (1)$$

The following proof is done by three steps: **1)** show that $p_M^{i,j}$ s is an optimal solution of the integer programming of the bipartite maximum weighted matching problem; **2)** show that any $p_R^{i,j}$ s is a feasible solution of the relaxed integer programming of the bipartite maximum weighted matching problem, which allows the integer variables to be relaxed as any real number between $[0, 1]$; **3)** Start from the fact that the achieved objective value of $p_M^{i,j}$ s in the integer programming is greater than or equal to that of $p_R^{i,j}$ s in the relaxed integer programming for the maximum weight matching problem [36] and show that Eq. (1) holds.

Step 1: Note that $p_M^{i,j}$ s are either 0 or 1, indicating whether to use client j as client i 's follower or not. First observe that since $r_j^{(i,j)}$ is greater than 0 for all $i, j \in V, i \neq j$, the size of the output of Problem 1, $|M|$, must be $|V|$. In other words, every client has a different follower, which implies

$$\sum_{j \in V \setminus \{i\}} p_M^{i,j} = 1, \forall i \in V. \quad (2)$$

Then, recall that Problem 1 is actually a bipartite maximum weighted maximum cardinality matching problem. The book [30] shows that, to solve the bipartite maximum weighted maximum cardinality matching problem, we can add a sufficiently large number, C , to the weight of each edge, and solve the bipartite maximum weighted matching problem on the new graph instead. We can now use the integer programming of the bipartite maximum weighted matching problem to find $p_M^{i,j}$ s. We first give a detailed construction of the bipartite graph. Given the set V and throughputs $r_j^{(i,j)}$ s in Problem 1, we construct a bipartite graph $G = (V_1 \cup V_2, E)$, where $V_1 = \{v_1^1, v_1^2, \dots, v_1^{|V|}\}$ (the set of winners of the first contention), $V_2 = \{v_2^1, v_2^2, \dots, v_2^{|V|}\}$ (the set of followers), $E = \{(v_1^i, v_2^j) | r_j^{(i,j)} > 0\}$, and $w(v_1^i, v_2^j) = r_j^{(i,j)} + C, \forall (v_1^i, v_2^j) \in E$ (the increased throughput).

The integer programming of the bipartite maximum weighted matching problem on G can then be formulated as

$$\begin{aligned} & \text{maximize} && \sum_{(v_1^i, v_2^j) \in E} w(v_1^i, v_2^j) p^{i,j} \\ & \text{subject to} && \sum_{j \in V \setminus \{i\}} p^{i,j} \leq 1, \forall i \in V, \end{aligned}$$

(the maximum number of edges incident to v_1^i is 1) Observe that

$$\sum_{i \in V \setminus \{j\}} p^{i,j} \leq 1, \forall j \in V,$$

(the maximum number of edges incident to v_2^j is 1) $p^{i,j} = 0$ or 1, $\forall i, j \in V, i \neq j$.

The optimal $p^{i,j}$ s of the above integer programming are then $p_M^{i,j}$ s.

Step 2: To show that $p_R^{i,j}$ s correspond to a feasible solution of the relaxed integer programming of the bipartite maximum weighted matching problem, we need to derive some properties of $p_R^{i,j}$ s. The fairness constraint requires that every client has the same probability to transmit the second stream. We hence have the following equation.

$$\sum_{i \in V \setminus \{j\}} p_R^{i,j} = \sum_{i \in V \setminus \{j'\}} p_R^{i,j'} \leq 1, \forall j, j' \in V. \quad (3)$$

The inequality must follow. Otherwise, $\sum_{i \in V} \sum_{j \in V \setminus \{i\}} p_R^{i,j} = \sum_{j \in V} \sum_{i \in V \setminus \{j\}} p_R^{i,j} > |V|$, which implies $\sum_{j \in V \setminus \{i\}} p_R^{i,j} > 1$ for some i and contradicts to the fact that any fair probabilistic assignment chooses at most one follower at each time, i.e.,

$$\sum_{j \in V \setminus \{i\}} p_R^{i,j} \leq 1, \forall i \in V. \quad (4)$$

Then, by Eq. (3) and Eq. (4), we get that $p_R^{i,j}$ s correspond to a feasible solution of the relaxed integer programming.

Step 3: It has been shown in [36] that, for the relaxed integer programming of the bipartite maximum weighted matching problem, the objective value achieved by an optimal integral solution is no less than that achieved by any feasible solution of the relaxed integer programming. Hence, we have $\sum_{(v_1^i, v_2^j) \in E} w(v_1^i, v_2^j) p_M^{i,j} \geq \sum_{(v_1^i, v_2^j) \in E} w(v_1^i, v_2^j) p_R^{i,j}$. Therefore, it is sufficient to show that

$$\sum_{(v_1^i, v_2^j) \in E} w(v_1^i, v_2^j) p_M^{i,j} \geq \sum_{(v_1^i, v_2^j) \in E} w(v_1^i, v_2^j) p_R^{i,j} \text{ implies Eq. (1).}$$

Before showing this, we must further refine Eq. (4) and get the following equation.

$$\sum_{j \in V \setminus \{i\}} p_R^{i,j} = 1, \forall i \in V, \quad (5)$$

which can be proved by contradiction. If there exists some $i' \in V$, such that $\sum_{j \in V \setminus \{i'\}} p_R^{i',j} < 1$, then we must have $\sum_{i \in V \setminus \{j\}} p_R^{i,j} = \sum_{i \in V \setminus \{j'\}} p_R^{i,j'} < 1, \forall j, j' \in V$; otherwise, if $\sum_{i \in V \setminus \{j\}} p_R^{i,j} = \sum_{i \in V \setminus \{j'\}} p_R^{i,j'} = 1, \forall j, j' \in V$, then $\sum_{j \in V \setminus \{i\}} p_R^{i,j} = 1, \forall i \in V$, which contradicts to $\sum_{j \in V \setminus \{i'\}} p_R^{i',j} < 1$. Therefore, we can add a small value to all $p_R^{i',j}$ s, $j \in V \setminus \{i'\}$, such that Eq. (3) and Eq. (4) still hold, i.e., after addition, $p_R^{i,j}$ s still correspond to a fair probabilistic assignment. Obviously, the average throughput is higher after we increase $p_R^{i',j}$ s, which contradicts to the fact that $p_R^{i,j}$ s correspond to an optimal fair probabilistic assignment. Hence, Eq. (5) holds.

We are now ready to accomplish the final part of the proof.

$$\begin{aligned} & \sum_{(v_1^i, v_2^j) \in E} w(v_1^i, v_2^j) p_M^{i,j} = \sum_{v_1^i \in V_1} \sum_{v_2^j \in V_2, j \neq i} w(v_1^i, v_2^j) p_M^{i,j} \\ & = \sum_{i \in V} \sum_{j \in V \setminus \{i\}} (r_j^{(i,j)} + C) p_M^{i,j} \end{aligned}$$

$$= \sum_{i \in V} \sum_{j \in V \setminus \{i\}} r_j^{(i,j)} p_M^{i,j} + C \sum_{i \in V} \sum_{j \in V \setminus \{i\}} p_M^{i,j},$$

and

$$\sum_{(v_1^i, v_2^j) \in E} w(v_1^i, v_2^j) p_R^{i,j} = \sum_{i \in V} \sum_{j \in V \setminus \{i\}} r_j^{(i,j)} p_R^{i,j} + C \sum_{i \in V} \sum_{j \in V \setminus \{i\}} p_R^{i,j}$$

by a similar reasoning. Therefore, it is sufficient to show that $\sum_{i \in V} \sum_{j \in V \setminus \{i\}} p_M^{i,j} = \sum_{i \in V} \sum_{j \in V \setminus \{i\}} p_R^{i,j}$. By Eq. (2) and Eq. (5), we have $\sum_{i \in V} \sum_{j \in V \setminus \{i\}} p_M^{i,j} = \sum_{i \in V} \sum_{j \in V \setminus \{i\}} p_R^{i,j} = |V|$. The proof is then completed.